Remarks

This preliminary amendment is filed for the purpose of placing the application into standard U.S. format and to correct any grammatical errors. Claims 2, 3, 9, 10, 11 and 16 have been amended. Consideration and allowance of the claims is earnestly solicited.

Attached hereto is a marked-up version of the changes made to the specification and claims by the current amendment. The attached page is captioned "Version with markings to show changes made."

Respectfully submitted,

Date: 1/22/02

Alfred A. Fressola, Reg. No. 27,550
Ware, Fressola, Van Der Sluys
& Adolphson LLP
Bradford Green, Building Five
755 Main Street, PO Box 224
Monroe, CT 06468
(203) 261-1234

AAF/aks

VERSION WITH MARKINGS TO SHOW CHANGES MADE

In the Specification:

Paragraph beginning at line 4 of page 4 has been amended as follows:

It is often necessary to standardize scattered documents and files that are in different formats in order to be able to further process them. The ability [Ability] to standardize is an important advantage when dealing with matters between separate enterprises, because in practice it is impossible to presuppose that all enterprises with mutual cooperation or with various client relationships should standardize their data systems and programs. In data processing, the ability to standardize facilitates not only the internal data processing of the enterprise, but especially the operation of the publishing and cooperation networks. It is essential to be able to select, among the electronic flow of information, those areas that are individually important, and to be able to observe, compare and further process said parts as a uniform entity.

Paragraph beginning at line 14 of page 4 has been amended as follows:

Figure 1 illustrates the basic principles of the invention. The code generation application is a tool arranged for separating, organizing and standardizing data. The code generation application comprises three parts: a code generation component 102, the generated extraction rules 103 and an extraction component 104. The source material 101 can be any IT-material, such as files, documents or continuous data stream. The only requirement for the source format is that immediately before or after the desired data field that should be extracted there must be provided a field separator having the length of one basic unit of the data to be processed, i.e. one token, and that this [said] field separator is repeated in all data records immediately before [said] the data field or immediately thereafter. However, the extracted record itself may contain various different field separators. The user selects either the whole source material or certain parts, i.e. data areas, therein. The data structure of the selected data areas must be processable as flat, i.e. it must not contain recursive structures. For instance a table or a list has this kind of flat data structure. If the user wishes to treat only a part of the structural data unit, for example a table, he must point out, as examples to the code generation component 102 of the code generation application, at least two such rows in the table that are mutually as different as possible. The more the examples differ from each other, the fewer examples there are needed in order to achieve the desired extraction result. This means that as many columns as possible in the chosen exemplary rows should contain different information.

In the Claims:

2. (Amended) A method according to claim 1, comprising the step of modifying the extracted data areas so as to be uniform in format.

3. (Amended) A method according to claim 1, wherein the at least two exemplary cases that are pointed out each have a structure and a content, and where the structure of each of the at least two exemplary cases is identical, but the content is different.

9. (Amended) A method according to claim 7, wherein in order to generate a set of rules, the method comprises the steps of:

- marking the longest of the selected, tokenized examples as a regular expression,
- marking the next longest of the selected, tokenized examples as an exemplary expression and
- comparing the regular expression with the exemplary expression [of the moment] in question.

10. (Amended) A method according to claim 9, wherein the regular expression and the exemplary expression [of the moment] in question are compared by means of a given reference algorithm that returns an edit script.

11. (Amended) A method according to claim 10, wherein the regular expression and the exemplary expression [of the moment] are compared by means of a reference algorithm that returns the shortest possible edit script.

16. (Amended) An arrangement according to claim 15, comprising means for modifying the extracted elements so as to be uniform in format.

In the Abstract:

Paragraph beginning at line 1 of the Abstract page has been amended as follows:

ABSTRACT OF THE DISCLOSURE

The invention relates to a method for generating rules, by which method data can be regrouped. In addition, the invention relates to an arrangement for realizing [said] this method. The object of the invention is to realize a method and arrangement whereby even a user without any skills in program-writing can generate extraction rules for data areas chosen from the source data. In [said] the method, the information contained in the original data is treated so that from the original source data, the user selects at least two exemplary cases. On the basis of the exemplary cases pointed out, there is generated a set of rules, and the data areas according to [said] these rules are extracted from the original source data. The extracted data areas can be further processed in a desired way.

[Figure 1]